

## **AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

- 1           1. (Currently amended) A method for facilitating typesafe software design  
2 while supporting structured composition of a software system, comprising:  
3           receiving a first invocation of the software system;  
4           assigning a first context to the first invocation;  
5           examining the first invocation to locate components of the first invocation;  
6           registering a unique factory to build each component in a software design  
7 environment, wherein the unique factory is ~~these factories are~~ registered using the  
8 first context;  
9           providing an additional factory for building an extended component of the  
10 first invocation; and  
11           when a component is needed, building the component using the unique  
12 factory associated with the component, whereby building the component after  
13 each component has a registered factory eliminates potential problems with  
14 initialization circularity.
- 1           2. (Currently amended) The method of claim 1, further comprising:  
2           receiving a second invocation of the software system;  
3           assigning a second context to the second invocation;  
4           examining the second invocation to locate components of the second  
5 invocation;

6        registering a unique factory to build each component in the software  
7        design environment, wherein the unique factory is ~~these factories are~~ registered  
8        using the second context; and  
9        when a component is needed, building the component using a factory  
10       associated with the component, whereby building the component after each  
11       component has a registered factory eliminates problems with initialization  
12       circularity.

1        3. (Original) The method of claim 2, wherein components from the second  
2        invocation are not available to the first invocation.

1        4 (Canceled).

1        5. (Currently amended) The method of claim 1, wherein registering the  
2        unique factory to build each component in the software design environment  
3        involves placing a key and a related factory identifier into a storage structure.

1        6. (Currently amended) The method of claim 5, wherein building the  
2        component in the software design environment using the unique factory ~~the factory~~  
3        associated with the component involves using the key to retrieve the related  
4        factory identifier from the storage structure.

1        7. (Original) The method of claim 6, wherein the storage structure includes  
2        a hash table.

1        8. (Currently amended) A computer-readable storage device storing  
2        instructions that when executed by a computer cause the computer to perform a

3 method for facilitating typesafe software design while supporting structured  
4 composition of a software system, the method comprising:  
5 receiving a first invocation of the software system;  
6 assigning a first context to the first invocation;  
7 examining the first invocation to locate components of the first invocation;  
8 registering a unique factory to build each component in a software design  
9 environment, wherein the unique factory is ~~these factories are~~ registered using the  
10 first context;  
11 providing an additional factory for building an extended component of the  
12 first invocation; and  
13 when a component is needed, building the component using the unique  
14 factory associated with the component, whereby building the component after  
15 each component has a registered factory eliminates potential problems with  
16 initialization circularity.

1 9. (Currently amended) The computer-readable storage device of claim 8,  
2 the method further comprising:  
3 receiving a second invocation of the software system;  
4 assigning a second context to the second invocation;  
5 examining the second invocation to locate components of the second  
6 invocation;  
7 registering a unique factory to build each component in the software  
8 design environment, wherein the unique factory is ~~these factories are~~ registered  
9 using the second context; and  
10 when a component is needed, building the component using a factory  
11 associated with the component, whereby building the component after each  
12 component has a registered factory eliminates problems with initialization  
13 circularity.

1           10. (Previously presented) The computer-readable storage device of claim  
2 9, wherein components from the second invocation are not available to the first  
3 invocation.

1           11 (Canceled).

1           12. (Currently amended) The computer-readable storage device of claim 8,  
2 wherein registering the unique factory to build each component in the software  
3 design environment involves placing a key and a related factory identifier into a  
4 storage structure.

1           13. (Currently amended) The computer-readable storage device of claim  
2 12, wherein building the component using the unique factory~~the factory~~  
3 associated with the component involves using the key to retrieve the related  
4 factory identifier from the storage structure.

1           14. (Previously presented) The computer-readable storage device of claim  
2 13, wherein the storage structure includes a hash table.

1           15-21 (Canceled).